

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

**Факультет інформатики та обчислювальної техніки
Кафедра автоматики та управління в технічних системах**

До захисту допущено:

Завідувач кафедри

_____ Олександр РОЛІК

«__» _____ 20__ р.

**Дипломний проект
на здобуття ступеня бакалавра
за освітньо-професійною програмою «Комп'ютеризовані системи управління»
спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології»
на тему: «Мобільна автономна система збору та сортування сміття»**

Виконав:

студент IV курсу, групи ІА-61

Ялинний Ігор Костянтинович _____

Керівник:

Заступник декана, к.т.н., доцент,

Писаренко Андрій Володимирович _____

Рецензент:

доц. каф. ТК, к.т.н., доц.

Ткач Михайло Мартинович _____

Засвідчую, що у цьому дипломному
проекті немає запозичень з праць інших
авторів без відповідних посилань.
Студент (-ка) _____

Київ – 2020 року

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Факультет інформатики та обчислювальної техніки

Автоматики та управління в технічних системах

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 151 «Автоматизація та комп'ютерно-інтегровані технології»

Освітньо-професійна програма «Комп'ютеризовані системи управління»

«Затверджую»

Завідувач кафедри

О.І.

Ролік

«__»

_20__р.

**ЗАВДАННЯ
на дипломний проєкт студенту
Ялинного Ігоря Костянтиновича**

1. Тема проєкту «Мобільна автономна система збору та сортування сміття», керівник проєкту Писаренко Андрій Володимирович к.т.н., доцент, затверджені наказом по університету від «_07_»__травня__2020р. №1081-с.

2. Термін подання студентом проєкту 09.06.2020

3. Вихідні дані до проєкту: Система, розмірами 370x240x100 мм, що здатна автоматично пересуватися по площині заданого маршруту, знаходити, підбирати та сортувати сміття, натренована модель комп'ютерного зору та програмне забезпечення для її використання.

4. Зміст пояснювальної записки: огляд існуючих рішень, опис розробленого рішення, опис структурної схеми, опис функціональної схеми, вибір окремих компонентів системи, опис алгоритму роботи системи, додаток для дистанційного керування системою.

5. Перелік графічного матеріалу: схема структурна, схема функціональна, блок-схема алгоритму роботи системи, блок-схема алгоритму розпізнавання сміття.

6. Дата видачі завдання 10 березня 2020
року _____

Календарний план

№	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1	Огляд існуючих рішень	10.03.20 – 17.03.20	
2	Створення вибірки зображень для тренування моделі комп'ютерного зору	20.03.20 – 01.04.20	
3	Встановлення програмного забезпечення для тренування моделі комп'ютерного зору	01.04.20 – 10.04.20	
4	Тренування моделі комп'ютерного зору	13.04.20 – 17.04.20	
5	Розробка додатку для використання моделі комп'ютерного зору	20.04.20 – 30.04.20	
6	Тестування додатку	29.04.20 – 01.05.20	
7	Проектування системи	01.05.20 – 10.05.20	
8	Опис спроектованої системи	10.05.20 – 20.05.20	
9	Розроблення схем	20.05.20 – 23.05.20	
10	Оформлення текстової документації	23.05.20 – 08.06.20	

Студент

Ігор ЯЛИННИЙ

Керівник

Андрій ПИСАРЕНКО

АНОТАЦІЯ

Ялинний І.К. Мобільна автономна система збирання сміття. КПІ ім. Ігоря Сікорського, Київ, 2020.

Пояснювальна записка складається з 9 розділів, 60 с. тексту, містить 25 рисунків, 1 таблицю, посилання на 43 літературних джерела, додатки та 4 кресленики.

Ключові слова: система збирання сміття, комп'ютерний зір, машинне навчання, розпізнавання сміття, автономна система.

Об'єктом розробки є мобільна автономна система збирання сміття.

Мета розробки – автоматизація процесу збирання вуличного сміття, а також економія людських ресурсів.

У дипломному проєкті розроблено автономну систему збирання сміття, що автоматично пересувається по заданій площині, за допомогою камери та комп'ютерного зору розпізнає сміття та збирає його маніпулятором у відповідний контейнер. Обрано мінімально необхідні компоненти, з яких можливо зібрати прототип системи, розрахована потужність необхідних акумуляторів, натреновано модель комп'ютерного зору, створено додаток для розпізнавання сміття.

Описана система є унікальною. На даний момент комерційних аналогів не існує.

SUMMARY

Yalynnyi I. K. Mobile autonomous system for garbage collection and sorting. Igor Sikorsky KPI, Kyiv, 2020.

The explanatory note consists of 9 sections, 60 pages of text, contains 25 figures, 1 table, references to 43 literature sources, appendices and 4 drawings.

Keywords: garbage collection system, computer vision, machine learning, garbage recognition, autonomous system.

The object of development is a mobile autonomous garbage collection system.

The purpose of the development is to automate the process of collecting street garbage, and also saving human resources.

The graduation project developed an autonomous garbage collection system that automatically moves on a given plane, with the help of a camera and computer vision recognizes garbage and collects it with a manipulator in the appropriate container. The minimum necessary components from which it is possible to assemble a prototype of the system are selected, the capacity of the necessary batteries is calculated, the model of computer vision is trained, the application for garbage recognition is created.

The described system is unique. At the moment there are no commercial analogues.

**Пояснювальна записка
до дипломного проекту
на тему: «Мобільна автономна система збору та
сортування сміття»**

Київ – 2020 року

ЗМІСТ

ВСТУП	10
1 ОГЛЯД ІСНУЮЧИХ РІШЕНЬ	11
1.1 Робот для автоматичного збирання сміття з трави	11
1.2 Робот для збирання сміття на пляжі	12
1.3 Автономний робот для збирання сміття.....	14
1.4 WasteShark	15
1.5 Висновки до розділу	16
2 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	18
3 ОПИС СТРУКТУРНОЇ СХЕМИ	20
4 ОПИС ФУНКЦІОНАЛЬНОЇ СХЕМИ.....	25
5 ВИБІР ОКРЕМИХ КОМПОНЕНТІВ СИСТЕМИ.....	28
6 ОПИС АЛГОРИТМУ РОБОТИ СИСТЕМИ.....	41
7 ДОДАТОК ДЛЯ ДИСТАНЦІЙНОГО КЕРУВАННЯ СИСТЕМОЮ	48
7.1 Підпрограма системи	48
7.2 Серверна частина	49
7.3 Мобільний додаток	49
8 РОЗРОБКА НЕЙРОМЕРЕЖІ РОЗПІЗНАВАННЯ СМІТТЯ.....	51
8.1 Тренування моделі комп'ютерного зору	53
8.2 Реалізація додатку для розпізнавання сміття.....	55

					ІА61.330БАК.005 ПЗ			
	Ар	№ докум.	Підп	Дат				
Розроб.	Ялинний				Мобільна автономна система збору та сортування сміття	Літ.	Арку	Аркуші
	Писарен						1	60
						НТУУ(КПІ) ФІОТ група ІА-61		
Н.								
Затв.								

ВИСНОВКИ59

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ61

					ІА61.330БАК.005 ПЗ	А
З	А	№ докум.	Підп	Д		8

ПЕРЕЛІК АБРЕВІАТУР, СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ

GPS – Global Pointing System

WSG84 – World Geodetic System 1984

LiDAR – Light Identification Detection and Ranging

ІВМ – Інерціальний вимірювальний модуль

МП – Мегапіксель

GPIO – General-purpose input/output

CSI – Camera serial interface

					ІА61.330БАК.005 ПЗ	А
З	А	№ докум.	Підп	Д		9

ВСТУП

Наразі є актуальною проблема забруднення вулиць сміттям. На його прибирання витрачається багато коштів та людських ресурсів, що з економічної точки зору невигідно.

Метою дипломного проєкту є автоматизація процесу збирання та сортування сміття за рахунок роботизованої системи. У даному дипломному проєкті були поставлені та вирішені наступні задачі:

- 1) полегшення пошуку сміття за рахунок комп'ютерного зору.
- 2) автоматизація збирання сміття, як наслідок – економія людських ресурсів.
- 3) сортування зібраного сміття: система здатна збирати сміття з різних типів матеріалів у різні контейнери, таким чином розділяти сміття, що можливо відправити на повторну переробку та сміття, а що необхідно утилізувати.

					ІА61.330БАК.005 ПЗ	10 ^А
З	А	№ докум.	Підп	Д		

1 ОГЛЯД ІСНУЮЧИХ РІШЕНЬ

1.1 Робот для автоматичного збирання сміття з трави

Даного робота розробили китайські інженери. Він використовує машинне навчання та здатен в реальному часі знаходити сміття, відрізнати його від інших предметів та збирати його. Даний робот зображений на рисунку 1.1.



Рисунок 1.1 – Робот для автоматичного збирання сміття з трави [1]

Склад прототипу:

- а) самокерований робот;
- б) датчики для сприйняття навколишнього середовища;
- в) контроллер;
- г) маніпулятор;
- д) контейнер для сміття;

					ІА61.330БАК.005 ПЗ	А ІІ
З	А	№ докум.	Підп	Д		

е) електродвигун.

Принцип роботи:

- 1) робот переміщується на колесах, що обертаються за допомогою електродвигуна, який виконує команди руху, отримані з контролера.
- 2) робот переміщується по маршруту. Для цього система користується інформацією зі спеціальної бази даних маршрутів та використовує інерціальний вимірювальний модуль, що включає в себе:
 - 2.1) GPS – для отримання координат місцезнаходження робота;
 - 2.2) трьохосьовий гіроскоп – для отримання інформації про кут нахилу робота відносно горизонту;
 - 2.3) трьохосьовий акселерометр – для отримання інформації щодо прискорення робота;
 - 2.4) цифровий процесор руху – для отримання інформації щодо переміщення робота у просторі.
- 3) для виявлення сміття використовується камера, що відправляє дані на контролер. На контролері в свою чергу є програмне забезпечення, що за допомогою попередньо натренованої моделі CNN розпізнає на зображенні з камери сміття.
- 4) після розпізнання сміття робот обчислює маршрут до об'єкту на основі даних з камери та ультразвукового датчику відстані, після чого під'їжджає до нього.
- 5) коли робот знаходиться на потрібній відстані перпендикулярно до сміття, за допомогою маніпулятора поміщає сміття у контейнер.

1.2 Робот для збирання сміття на пляжі

Цей винахід інженерів з Сінгапуру створений для очистки піщаних пляжів від сміття. Продемонстрований на рисунку 1.2.

					ІА61.330БАК.005 ПЗ	12 ^А
З	А	№ докум.	Підп	Д		



Рисунок 1.2 – Робот для збирання сміття на пляжі [2]

Склад прототипу:

- а) робот;
- б) сонячна батарея;
- в) ківш;
- г) контейнер для сміття;
- д) мікроконтролер;
- е) модуль бездротового зв'язку Bluetooth;
- є) бездротова IP камера.

Принцип роботи:

- 1) керування роботом відбувається користувачем за допомогою комп'ютера через бездротовий зв'язок. Для цього до мікроконтролера під'єднаний модуль Bluetooth.
- 2) робот пересувається по піску за допомогою гусеничної системи та транслює зображення з камери користувачу.

- 3) коли робот отримує команду зібрати сміття, він переносить вміст ківша у контейнер, де сміття відфільтровується від піску.
- 4) на верхній частині робота знаходиться сонячна батарея, що заряджає акумулятор.

1.3 Автономний робот для збирання сміття

Даний робот є розробкою інженерів з Індії. На даний момент він знаходиться на стадії прототипу. Метою створення даного концепту було очищення вулиць від сміття. Робот працює автономно, не потребує людського втручання.

Склад прототипу:

- металевий каркас 50x40 мм;
- електричний двигун;
- акумулятор потужністю 12В та ємністю 12000 мАг;
- мікроконтролер Arduino Uno;
- три датчики відстані;
- драйвери електродвигунів;
- шасі з колесами;
- механізм збирання сміття;
- контейнер для сміття.

Принцип роботи:

- 1) після вмикання робота він починає пересуватися вперед та вмикається механізм збирання сміття, що знаходиться на передній частині робота. Механізм збирання сміття складається з чотирьох плоских прямокутних пластин, що закріплені під кутом 90° одна до одної та електродвигуна, що обертає їх навколо осі.

					ІА61.330БАК.005 ПЗ	14 ^А
З	А	№ докум.	Підп	Д		

- 2) коли на шляху робота трапляється сміття висотою 5 – 20 см, одна з плоских частин механізму, що обертається, заштовхує сміття у контейнер, що знаходиться за механізмом.
- 3) якщо один із датчиків відстані реєструє перешкоду, контролер віддає команду повороту і система повертає до тих пір, поки датчик не перестане реєструвати перешкоду.

Описаний робот продемонстрований на рисунку 1.3.

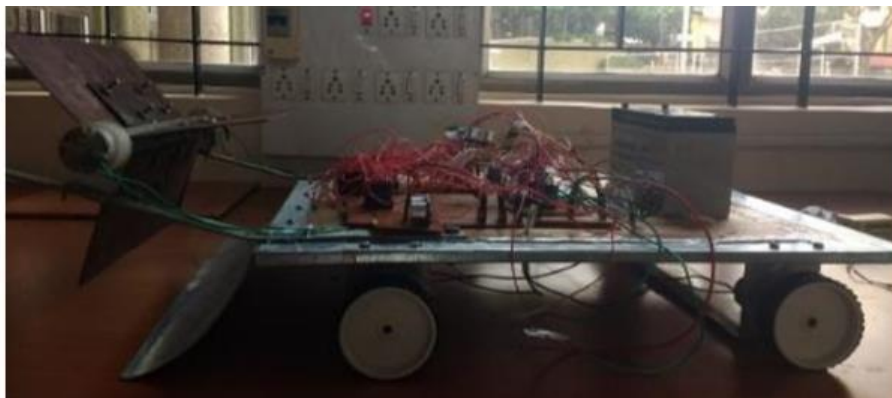


Рисунок 1.3 - Автономний робот для збирання сміття [40]

1.4 WasteShark

Дане рішення призначене для автономного плавання у гаванях та очищення поверхні води від відходів то того, як вони попадуть у море. WasteShark виготовлений голандською компанією Hardiman RanMarine та наразі тестується у водних гаванях Нідерландів.

Система має форму катамарану, конструкція складається з двох плавучих корпусів, що частково занурюються у воду, а також із основного корпусу, що містить у собі електродвигуни, акумулятори, контейнер для сміття, а також механізм з ківшами для збирання сміття з водної поверхні.

У майбутньому планується оснастити WasteShark датчиками для контролю якості та температури води, а також створити більшу версію із

					ІА61.330БАК.005 ПЗ	15 ^А
З	А	№ докум.	Підп	Д		

сонячними батареями, що буде вміщати до 500 кг сміття. Дане рішення продемонстровано на рисунку 1.4.



Рисунок 1.4 – WasteShark [42]

1.5 Висновки до розділу

У ході огляду існуючих рішень було виявлено, що на даний момент існує система для автоматичного прибирання сміття, що використовує комп'ютерний зір для його пошуку. Дана система здатна пересуватися по таким поверхням, як трава та точно розрізняти велику кількість різноманітного сміття. Недоліками даної системи є відсутність алгоритму для оптимального проходження площини маршруту, не досить надійна ходова частина, а також неможливість автоматичного сортування зібраного сміття.

Також було розглянуте рішення для збирання сміття на піщаних пляжах. Дане рішення має такі переваги, як сонячна батарея, завдяки якій акумулятор

					ІА61.330БАК.005 ПЗ	16 ^А
З	А	№ докум.	Підп	Д		

підзаряджається під час роботи системи. Серед недоліків системи є неможливість автономної роботи без втручання людини – система керується людиною, що отримує зображення з камери та керує процесом збирання сміття. Також недоліком є неможливість автоматичного сортування сміття – усе зібране поміщається у єдину ємність.

Було розглянуте рішення для очищення водних гаваней від сміття. Дане рішення має такі переваги, як повна автономність, захищеність від зовнішніх факторів, наявність працюючого прототипу.

На основі аналізу описаних аналогів, можна зробити висновок, що наразі не існує систем автоматичного збирання та сортування сміття із алгоритмами проходження заданої площини маршруту, що здатні працювати на вулицях міста та бездоріжжі без втручання людини.

					ІА61.330БАК.005 ПЗ	17 ^А
З	А	№ докум.	Підп	Д		

2 Аналіз предметної області

Оскільки мета дипломного проєкту полягає у розробці мобільної автономної системи збору та сортування сміття, після аналізу переваг та недоліків існуючих рішень, було встановлено, що дана система повинна:

- виконувати основну частину роботи без людського впливу;
- ефективно відрізняти сміття від інших об'єктів та відносити до відповідної категорії;
- автоматично будувати маршрут з урахуванням перешкод;
- пересуватися в умовах складної поверхні;
- дистанційно відстежуватися;
- мати можливість повернення за командою користувача.

Для досягнення поставлених цілей система повинна складатися з наступних компонентів:

- шасі робота – для механічного пересування системи;
- електродвигунів – для надання обертів колесам і, як наслідок, приведення системи у стан руху;
- акумуляторних батарей – для забезпечення струмом компонентів, що цього потребують;
- контролерів живлення – для забезпечення стабільної роботи акумуляторів;
- зарядного порту – для заряджання акумуляторів;
- камери – для можливості розпізнавання сміття перед системою;
- датчику LiDAR – для отримання відстані до знайденого сміття;
- інерціального вимірювального модуля – для отримання необхідних даних щодо знаходження системи у навколишньому середовищі;
- модуля зв'язку – для зв'язку системи із сервером та отримання координат місцезнаходження;

					ІА61.330БАК.005 ПЗ	18 ^А
З	А	№ докум.	Підп	Д		

- контролеру – для обробки зібраних даних та керування пристроями;
- маніпулятора – для механічного збору сміття;
- драйвера електродвигунів – для підключення електродвигунів до контролеру та акумуляторів;
- драйвера сервоприводів – для підключення сервоприводів до контролеру та акумуляторів;
- модуля пам'яті – для зберігання даних, необхідних для роботи системи;
- двох контейнерів для сміття – для окремого зберігання сміття, що можна переробити та сміття, що підлягає утилізації.

Для розпізнання сміття від інших об'єктів необхідно використати технології машинного навчання та комп'ютерного зору.

					ІА61.330БАК.005 ПЗ	19 ^А
З	А	№ докум.	Підп	Д		

3 ОПИС СТРУКТУРНОЇ СХЕМИ

Структурна схема системи представлена на кресленику ІА61.330БАК.005 Э1.

Блок модулю навігації відповідає за орієнтування робота у місцевості, обчислення його переміщення, швидкості та необхідних даних для побудови маршруту руху та розпізнавання та сортування сміття.

Даний блок складається із п'яти основних частин: одометр, ІВМ, камера, модуль зв'язку та сенсор LiDAR. Кожна з цих частин – пристрій для отримання інформації щодо навколишнього середовища та актуального місцезнаходження для подальшої відправки зібраної інформації на контролер.

Одометр – це пристрій для вимірювання кількості обертів колеса. Використовується для обчислення пройденого шляху. Одометр у даній системі складається із датчика Холла (рисунок 3.1), що закріплюється на статорі електродвигуна робота та вимірює напруженість магнітного поля, та мікроконтролеру, що збирає дані з датчику та відправляє на контролер.

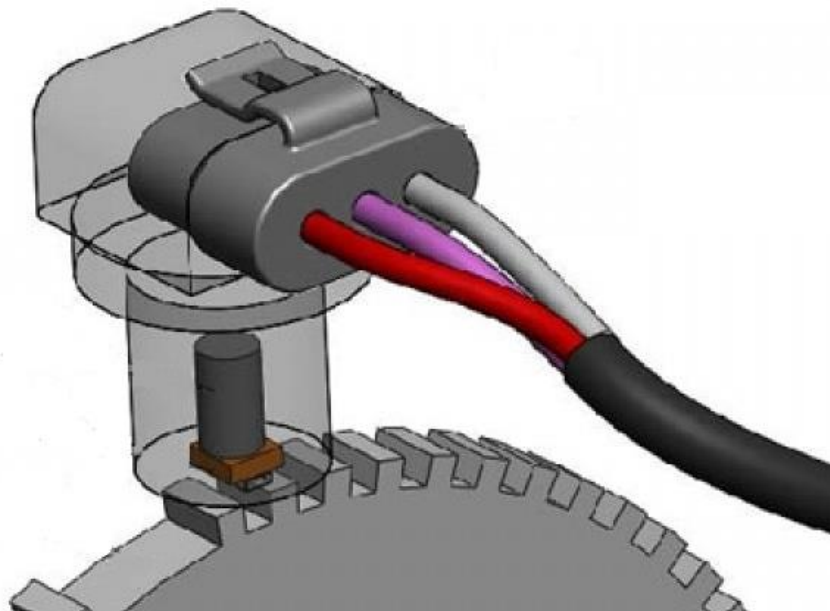


Рисунок 3.1 – Датчик Холла [4]

					ІА61.330БАК.005 ПЗ	А 20
З	А	№ докум.	Підп	Д		

ІВМ складається з комплексу датчиків, що потрібні для вимірювання різних змінних положення системи у просторі та змінних навколишнього середовища.

У даній системі ІВМ потрібен для відправлення зібраної інформації на контролер, де вона використовується для обчислень щодо переміщення робота та його маршруту.

Камера необхідна для отримання зображення перед роботом з ціллю пошуку сміття за допомогою комп'ютерного зору та сортування його у відповідну категорію. Камера повинна бути закріплена на передній частині робота та відправляти зображення в реальному часі на контролер.

Модуль зв'язку містить у собі два основні компоненти:

а) LTE модуль – модуль зв'язку для підключення до мережі Інтернет. Потрібен в системі для дистанційного приймання команд від користувача, а також для відправлення користувачу координат, отриманих с GPS модуля з метою відстеження місцезнаходження системи у реальному часі.

б) GPS модуль – це модуль супутникової навігації, що потрібен в системі для отримання точного положення у просторі у всесвітній системі координат WSG84 з метою побудови маршруту, використовуючи базу даних з мапою місцевості.

Сенсор LiDAR – пристрій, що використовує сигнали відбитого від поверхні об'єктів світла для отримання інформації про об'єкти, що знаходяться на певній відстані. Такі сенсори широко використовуються для машин з автопілотом для отримання інформації щодо інших учасників руху та перешкод.

Даний сенсор повинен бути розташований поряд із камерою, щоб інформацію, отриману з нього можна було зіставити із зображенням,

					ІА61.330БАК.005 ПЗ	А 21
З	А	№ докум.	Підп	Д		

отриманим з камери. Таким чином можна отримати відстань до об'єкту, що розпізнала камера, а також до перешкод, що потрібно оминати.

Модуль переміщення необхідний для переміщення системи у необхідне місце. Даний модуль складається із двох основних частин: електродвигун та ходова частина.

Електродвигун в системі необхідний для перетворення електричної енергії у механічну з метою надання обертів колесам ходової частини і, як наслідок, руху системи.

Принцип дії електродвигуна базується на електромагнітній взаємодії частин статора на частини ротора, і як наслідок, обертання ротора навколо осі.

Ходова частина – конструкція, що об'єднує в собі:

- ведучі та неведучі колеса – для пересування системи по землі;
- зубчасті механізми, що взаємодіють із ротором двигуна, завдяки чому відбувається передавання моменту обертання на ведучі колеса спроектованого робота;
- шасі – частина, що утримує колеса, електромотор, зубчасті механізми на місці, а також містить в собі механізм для повороту.

У даній системі ходова частина використовує принцип однобортного управління, що продемонстрований на рисунку 3.2. Такий принцип управління, із використанням гусеничного шасі, має наступні переваги для даної системи:

- система може виконувати поворот нульового радіусу, що надає можливість розвертання на місці;
- надає змогу більш точно оминати перешкоди, що трапляються на шляху пересування робота.
- можливість пересуватися по поверхні з нерівностями, піщаним пляжам тощо.

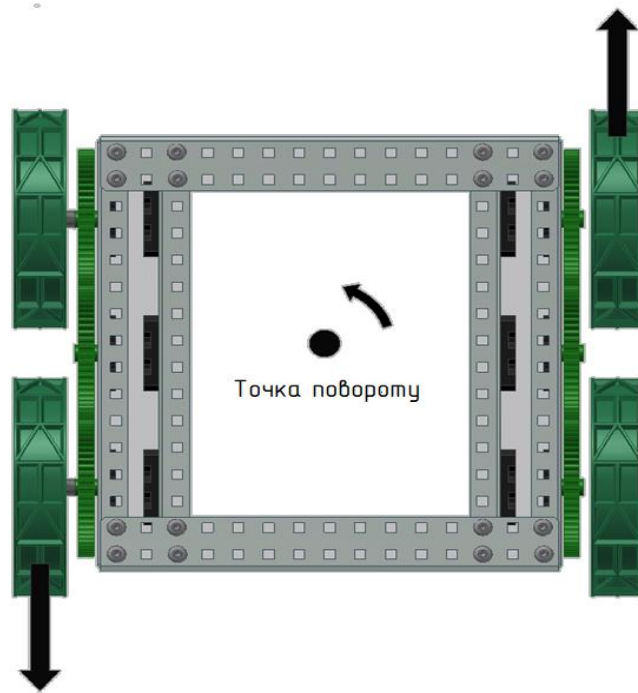


Рисунок 3.2 – Принцип однобортного управління [9]

Модуль живлення – блок, що відповідає за живлення електроенергією усіх електричних компонентів системи та зберігання електричного заряду для тривалої автономної роботи системи.

Акумулятор необхідний для зберігання електричного заряду для автономної роботи системи без необхідності підключення до мережі живлення.

Контролер живлення – пристрій, що відповідає за зарядження акумулятора, регулювання сили струму та напруги, а також за живлення струмом компонентів системи, що цього потребують.

Зарядний порт – це інтерфейс, що з'єднується із мережею живлення через спеціальний кабель та подає електричний струм на контролер живлення для заряджання акумулятора.

Блок «Модуль збирання сміття» відповідає за механічне збирання, сортування та зберігання сміття. Він складається із двох основних частин: маніпулятора та контейнерів для сміття.

Маніпулятор – це механізм, що призначений для роботизованого піднімання об’єктів та їх складання в іншу точку. В даному випадку маніпулятор повинен піднімати об’єкт перед системою, що розпізнався, як сміття одного з двох наступних типів: сміття, що можна переробити та сміття, що не можна переробити та перенести у відповідний контейнер.

На корпусі системи встановлено два контейнери для різних типів сміття. Після їх наповнення вони відкріплюються від корпусу та вміст контейнерів спорожняється.

Контролер складається із основних частин, таких як:

- процесор – для обчислень програм, що виконуються;
- оперативна пам’ять – пам’ять із великою швидкістю для тимчасових даних програм, до яких потрібен швидкий доступ;
- постійна пам’ять – пам’ять, у якій зберігаються програми, що працюють на контролері, а також база даних з картою місцевості.
- материнська плата – плата із мікросхемами та компонентами, що об’єднує усі інші компоненти контролера, а також має інтерфейси, необхідні для підключення до контролера компонентів системи збору інформації та виконання певних дій.

Контролер необхідний для обробки зібраних даних та керування системою на їх основі.

4 ОПИС ФУНКЦІОНАЛЬНОЇ СХЕМИ

Функціональна схема системи представлена на кресленику ІА61.330БАК.005
Э2.

На функціональній схемі представлена система, центром якої виступає контролер. Контролер у даній системі збирає інформацію з датчиків, обробляє її, а також керує підключеними до нього пристроями. Контролер має наступні інтерфейси:

- а) порт USB Type C для живлення контролеру. Через нього контролер живиться від акумуляторних батарей;
- б) CSI порт для підключення камери;
- в) 40-піновий інтерфейс GPIO – через цей інтерфейс до контролеру підключаються різні пристрої вводу та виводу інформації;
- г) інтерфейс для підключення картки пам'яті MicroSD.

Також на контролері присутні інші інтерфейси, такі, як USB, та MiniJack, але у даній системі вони не використовуються.

У блоці «Модуль навігації» представлені пристрої збору інформації для роботи системи. Усі вони підключені до інтерфейсів контролера для можливості відправки на нього зібрану інформацію. Серед пристроїв цього блоку:

- а) одометр – підключений через інтерфейс GPIO до контролера. Відправляє швидкість обертів колеса на контролер.
- б) гіроскоп – підключений через інтерфейс GPIO та передає інформацію щодо нахилу системи у просторі;
- в) акселерометр – пристрій для відправки даних щодо прискорення системи, підключений через інтерфейс GPIO;

					ІА61.330БАК.005 ПЗ	А 25
З	А	№ докум.	Підп	Д		

г) датчик температури – це датчик, що підключено до інтерфейсу GPIO, через який відправляє дані щодо поточної температури навколишнього середовища;

г) камера – підключена через інтерфейс CSI, для відправлення потокового зображення. Даний інтерфейс має велику швидкість передачі даних для стабільної передачі потокового відео, через що він є найбільш придатним для підключення камери;

д) модуль зв'язку, що включає у себе модуль сотового зв'язку для підключення до мережі Інтернет, а також GPS-модуль, що підключається до портів GPIO контролера для передачі інформації через сервер до користувача;

е) сенсор LiDAR – підключений до порту GPIO, через який передає інформацію щодо відстані до об'єкту.

Окремим пристроєм є модуль пам'яті, підключений через інтерфейс MicroSD, що використовується для зберігання та зчитування такої інформації, як:

- операційна система контролеру;
- програмне забезпечення системи;
- база даних із місцевістю, де повинна працювати система.

У блоці «Модуль живлення» зображені компоненти системи, що відповідають за живлення інших компонентів системи. Сполучною ланкою цього блоку є контролер живлення, що поєднує акумуляторні батареї із зарядним портом та компонентами системи, що потребують живлення. Контролер живлення регулює силу струму, що подається з зарядного порту на акумулятори, запобігаючи короткому замиканню, а також регулює струм, що подається на компоненти.

У блоці «Модуль переміщення» зображені компоненти, які відповідають за пересування системи. Частиною, що пов'язує компоненти цього блоку з іншими є драйвер. Драйвер має дві пари входів, до яких підключений

контролер через інтерфейс GPIO, та контролер живлення для живлення двигунів струмом, а також пару виходів, які підключені до двигунів. Даний драйвер приймає керуючі сигнали від контролеру та використовує струм з акумуляторів для керування двигунами. Двигуни в свою чергу з'єднані із ведучими колесами.

Блок «Модуль збирання сміття» містить в собі маніпулятор із драйвером, що підключений до сервоприводів маніпулятора та до контролеру з акумулятором аналогічно до драйверу електродвигунів. Також цей блок містить контейнери для сміття. Контейнери розміщені статично відносно маніпулятора, тож положення, у якому маніпулятор складає сміття у той, чи інший контейнер – стає.

					ІА61.330БАК.005 ПЗ	27 ^А
З	А	№ докум.	Підп	Д		

5 ВИБІР ОКРЕМИХ КОМПОНЕНТІВ СИСТЕМИ

5.1 Контролер

Обираючи контролер, треба враховувати такі фактори, як:

- обчислювальна потужність – вона повинна бути достатньою, щоб в реальному часі обробляти потокове зображення з камери, використовуючи модель комп'ютерного зору, обробляти дані, отримані з датчиків та давати команди на виконавчі пристрої;
- розмір – контролер не повинен займати велику кількість місця на корпусі;
- контролер повинен мати достатню кількість портів для підключення необхідних компонентів.

Оглянувши існуючі на сьогоднішній день контролери, було зроблено висновок, що найкращим варіантом буде Raspberry Pi 4 Model B від компанії Raspberry (рисунок 5.1), так, як контролери від цієї компанії є надійними, мають достатню кількість інтерфейсів для підключення пристроїв керування, а також мають достатню обчислювальну потужність для роботи з моделлю комп'ютерного зору. Обрана модель Raspberry Pi – найпотужніша на сьогоднішній день, та має усі необхідні інтерфейси. Також значною перевагою даного контролеру є операційна система Raspbian OS, особливостями якої у порівнянні із операційними системами інших контролерів є можливість зручного програмування високорівневими мовами, а також налаштування різних параметрів роботи контролера.

					ІА61.330БАК.005 ПЗ	28 ^А
З	А	№ докум.	Підп	Д		



Рисунок 5.1 – Raspberry pi 4 model b [14]

Обраний контролер має наступні важливі для системи технічні характеристики:

- а) чотирьох'ядерний процесор Broadcom BCM2711 із тактовою частотою 1.5 ГГц;
- б) 4 Гб оперативної пам'яті;
- в) відеоадаптер VideoCore VI;
- г) інтерфейс для підключення карти пам'яті формату MicroSD;
- д) інтерфейс для підключення камери;
- е) 40-піновий інтерфейс GPIO для підключення різних компонентів системи;
- є) порт USB Type C для живлення контролера.

Отже, обчислювальної потужності обраного контролера достатньо для швидкої роботи комп'ютерного зору та керування підключеними компонентами, а кількість інтерфейсів достатня для їх підключення.

5.2 Камера

Обираючи камеру для даної системи необхідно враховувати такі параметри:

- а) зображення з камери повинно мати роздільну здатність як мінімум 5 МП, щоб комп'ютерний зір був здатен розрізнити об'єкти на кадрі;
- б) кут огляду камери повинен бути як мінімум 120° для того, щоб робот міг знаходити сміття перед собою по всій ширині маршруту;
- в) камера повинна мати інфрачервону підсвітку для того, щоб мати можливість працювати при поганому освітленні, або вночі;
- д) камера повинна мати інтерфейс підключення CSI до обраного контролера.

Після огляду існуючих варіантів, для системи була обрана камера BauTech для Raspberry pi (рисунок 5.2).



Рисунок 5.2 – Камера BauTech для Raspberry pi [16]

					ІА61.330БАК.005 ПЗ	30 ^А
З	А	№ докум.	Підп	Д		

Обрана камера має наступні основні характеристики, що важливі для системи:

- а) матриця з роздільною здатністю 5 МП;
- б) кут огляду 160°;
- в) інфрачервону підсвітку для нічної з'йомки.

Отже, обрана камера підходить під вимоги та може бути використана в даній системі.

5.3 Інерціальний вимірювальний модуль

Для системи було обрано модуль від компанії Analog Devices, а саме модель ADIS16500, що продемонстрована на рисунку 5.3.

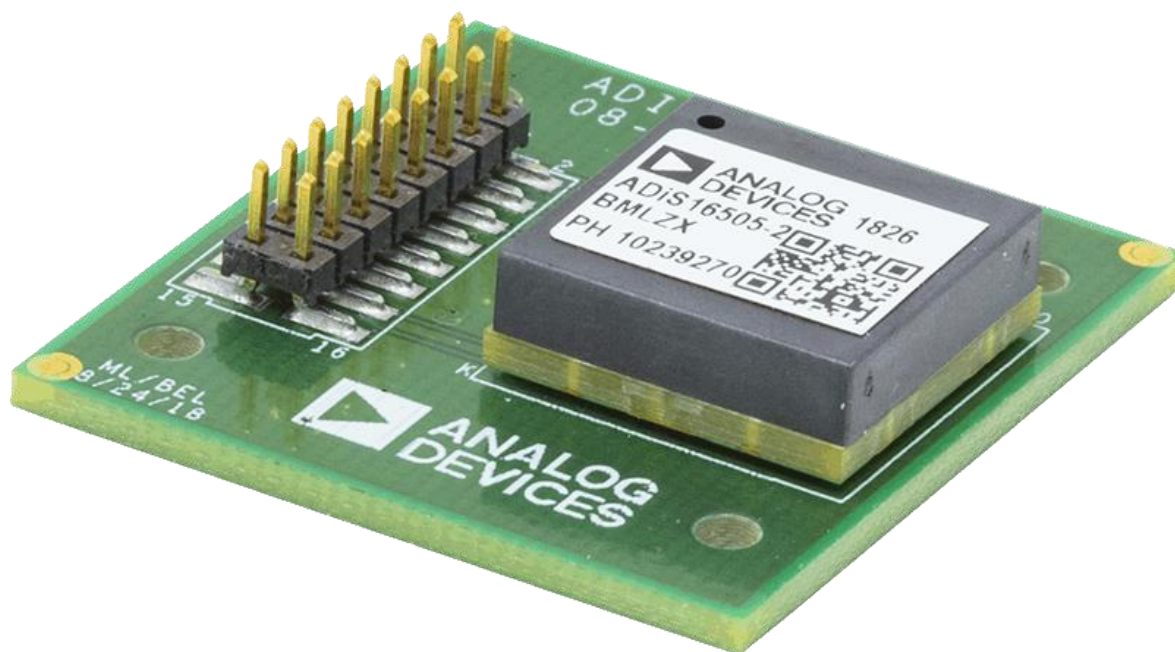


Рисунок 5.3 – Інерціальний вимірювальний модуль ADIS16500 [18]

Дана модель була обрана через високу точність вимірювання та наявність необхідних датчиків. А саме:

					ІА61.330БАК.005 ПЗ	31 ^А
З	А	№ докум.	Підп	Д		

- а) трьохосьовий цифровий гіроскоп – для вимірювання зміни кутів орієнтації системи у просторі;
- б) акселерометр – для вимірювання прискорення системи;
- в) датчик температури – для вимірювання температури навколишнього середовища та оцінки рівня безпеки для роботи системи.

5.4 Модуль LiDAR

Для даної системи було обрано датчик від компанії RCloud, а саме модель TF-Luna (рисунок 5.4).



Рисунок 5.4 – LiDAR RCloud TF-Luna [20]

Даний датчик має робочий діапазон від 0,2 до 8 метрів, регулює частоту оновлення кадрів до 250 Гц, кут огляду 2°, а напругу живлення – 5В. TF-Luna можна підключити до інтерфейсу GPIO контролера.

					ІА61.330БАК.005 ПЗ	32 ^А
З	А	№ докум.	Підп	Д		

5.5 Модуль зв'язку

Для системи було обрано модуль зв'язку для Raspberry pi від компанії Waveshare.

На рисунку 5.5 зображено модуль зв'язку, встановлений на контролер Raspberry pi.



Рисунок 5.5 – Модуль зв'язку [22]

Даний модуль зв'язку підтримує стандарт LTE CAT4 зі швидкістю передачі до 150 Мбіт/с для швидкої передачі даних при доволі низькому енергоспоживанні – 5В. Також наявна підтримка технології знаходження місцеположення – GPS із точністю визначення до 2.5 м.

Цей модуль має інтерфейс для підключення до пінів GPIO, що займає усі 40 пінів, а також на платі є власні 40 пінів GPIO.

5.6 Ходова частина та електродвигун

Наразі існує багато варіантів ходової частини для роботів. Найкращим варіантом є ходова частина, яку можна підключити до обраного контролера.

Після вивчення існуючих варіантів було обрано ходову частину RC Robot (рисунок 5.6).

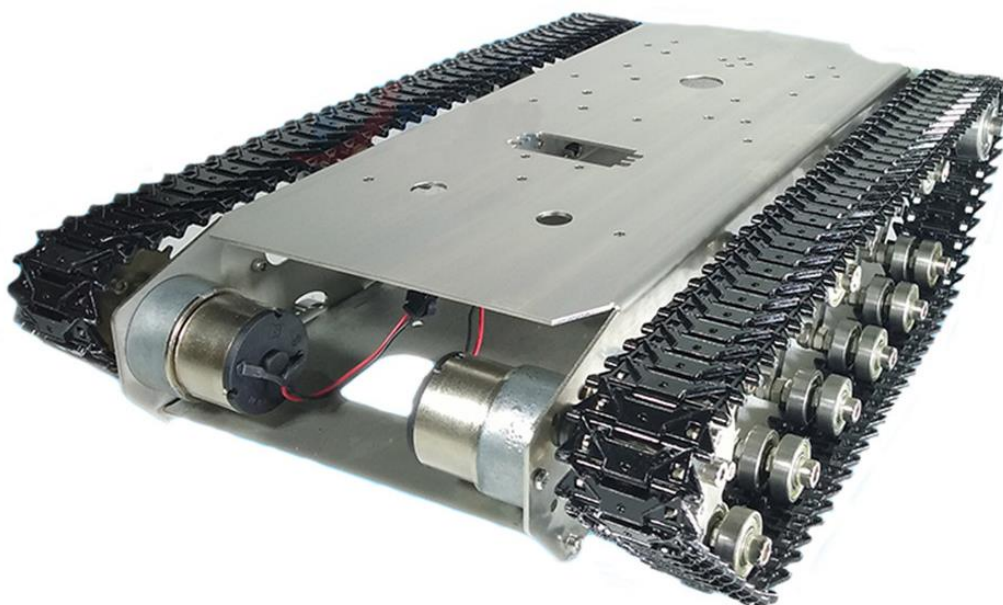


Рисунок 5.6 – Ходова частина RC Robot [24]

Дана ходова частина має гусеничну ходу, що дає можливість системі працювати у складних умовах, а також виконувати поворот нульового радіусу, завдяки чому система може розвертатись на місці.

Також перевагою цієї ходової частини є те, що вона вже містить в собі два електродвигуни, що під'єднані до лівої та правої гусеничної системи, а саме до її ведучих коліс.

Ходова частина для роботи потребує напругу від 6 до 18В та силу струму 12А. Ведучі колеса здатні розвивати швидкість до 13000 обертів за хвилину, тож потужності достатньо для пересування системи разом із вагою контейнерів зі сміттям. Вага ходової частини із двигунами – 4.1кг.

5.7 Маніпулятор

Після детального вивчення існуючих варіантів було обрано маніпулятор DIY 6 DOF (рисунок 5.7).



Рисунок 5.7 – Маніпулятор DIY 6 DOF [26]

Даний маніпулятор має кріплення, завдяки яким його можна закріпити на ходовій частині системи, а також інтерфейс живлення, що підключається

					ІА61.330БАК.005 ПЗ	35 ^А
З	А	№ докум.	Підп	Д		

до акумулятора та інтерфейс GPIO, завдяки якому маніпулятор підключається до обраного контролера.

5.8 Джерело живлення

Обираючи джерело живлення для системи, необхідно враховувати потужність, що необхідна для живлення усіх її компонентів. Тож перед тим, як робити вибір, необхідно розрахувати сумарну потужність усіх компонентів.

Для того, щоб знайти потужність, необхідну для живлення одного елементу, необхідно використати формулу 5.1

$$P = U \cdot I, \quad (5.1)$$

де P – потужність, необхідна для живлення компоненту;

U – необхідна напруга;

I – необхідна сила струму.

Двигуни ходової системи на піковій потужності потребують 18В, 12А.

Тож, розрахувавши потужність за вищенаведеною формулою, було отримано 216 Вт. Аналогічним чином було розраховано потужність кожного компонента системи та отримано дані, наведені у таблиці 5.1.

Таблиця 5.1 – необхідна потужність для живлення компонентів

Компонент	Необхідна потужність
Електродвигуни	216 Вт
Контролер	15 Вт
Маніпулятор	30 Вт
ІВМ	1.8 Вт
Камера	Живиться від контролера

LiDAR	0.35 Вт
-------	---------

Обчисливши потужності, необхідні для живлення усіх компонентів, тепер необхідно знайти суму потужностей для вибору акумулятора, вона складає 264 Вт.

Для системи було обрано літій-іонні високоструміві акумулятори Samsung INR18650 (рисунок 5.8). Причиною для такого вибору стало те, що акумулятори такого типу відмінно себе зарекомендували у сучасних електромобілях, де їх підключають у великій кількості блоками, а модель від компанії Samsung – дешевша за більшість аналогів, але з високим рівнем якості.



Рисунок 5.8 – Акумуляторні батареї Samsung INR18650 [28]

Номінальна потужність такого акумулятора – 36.5 Ватт, тож для забезпечення необхідної потужності, необхідно під'єднати певну кількість акумуляторів послідовно, щоб потужність такого блоку дорівнювала сумі

					IA61.330БАК.005 ПЗ	37 ^А
З	А	№ докум.	Підп	Д		

потужностей акумуляторів. Таким чином, можна порахувати, що для системи знадобиться 8 акумуляторів.

Розмір однієї акумуляторної батареї – 65мм в довжину та 18.4мм в діаметрі. Якщо розташувати усі батареї послідовно – цей блок досягне розміру у 520мм в довжину, що на 50мм більше за довжину корпусу системи, в ролі якої виступає її ходова частина. Рішенням цієї проблеми стало розміщення акумуляторів паралельно із послідовним підключенням за допомогою восьми батарейних блоків .

Вісім під'єднаних послідовно акумуляторів, розташованих паралельно у блоках займають 147.2 мм в ширину при ширині корпусу 150мм.

Отримано блок акумуляторних батарей із потужністю 292 Ватт та ємністю 16000 мАг.

5.9 Контролер живлення

Контролер живлення необхідний у системі, де є акумуляторні батареї для правильної подачі напруги при заряджанні батарей, уникнення коротких замикань і, як наслідок, подовшення строку експлуатації батарей.

Для батарей типу 18650 часто використовують контролери BMS. Тому було обрано контролер BMS 3s. Один такий контролер розрахований на 3 акумуляторних батареї. Тому для 8 батарей знадобиться 3 контролери. Контролер живлення продемонстровано на рисунку 5.9.

					ІА61.330БАК.005 ПЗ	38 ^А
З	А	№ докум.	Підп	Д		

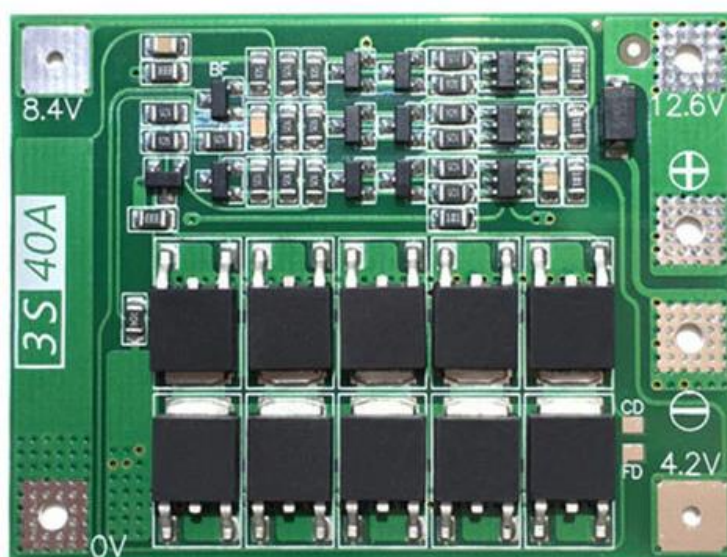


Рисунок 5.9 – Контролер живлення BMS 3s [30]

5.10 Модуль заряджання

Модуль заряджання необхідний для підключення інтерфейсу зарядки та заряджання акумуляторних батарей.

Для даної системи було обрано модуль заряджання TP 4056, що зображено на рисунку 5.10.



Рисунок 5.10 – Модуль заряджання акумулятора [33]

5.11 Драйвер електродвигуна

Драйвер – це пристрій для підключення до контролер компонентів, що потребують додаткового живлення, таких, як електродвигуни.

Обраний драйвер L298N MINI (рисунок 5.11) має дві пари входів для підключення обраного контролера та живлення, а також дві пари виходів для можливості підключення двох електродвигунів.



Рисунок 5.11 – Драйвер електродвигуна L298N MINI [35]

					ІА61.330БАК.005 ПЗ	40 ^А
З	А	№ докум.	Підп	Д		

6 ОПИС АЛГОРИТМУ РОБОТИ СИСТЕМИ

Загальний алгоритм роботи системи представлено на блок-схемі ІА61.330БАК.005 Д1.

Блок 2. Рішення «Чи була команда повернення». На даному етапі контролер за допомогою модуля зв'язку відправляє на сервер запит про наявність команди повернення на початкові координати від користувача. Якщо сервер повертає негативний результат – команди на повернення не було та система продовжує роботу. Якщо ж результат позитивний – це означає, що користувач відправив команду повернення системи на початкові координати, тож рішення системи в такому випадку – повернення на початкові координати.

Блок 3. Побудова маршруту. На даному етапі контролер обчислює маршрут для пошуку сміття. Для цього із модуля пам'яті програма на контролері використовує базу даних із мапою місцевості та координати площини місцевості, в якій необхідно прибрати сміття. Також використовується модуль GPS для отримання поточних координат у якості точки, з якої необхідно будувати маршрут.

Для того, щоб система виконала роботу у заданій площині, прибравши максимальну кількість сміття, необхідний алгоритм побудови шляху для проходження максимальної площі з урахуванням перешкод. Для цього алгоритм можна розділити на 2 основні етапи:

- а) побудова маршруту проходження площини без урахування перешкод;
- б) алгоритм хвильового трасування.

Для того, щоб побудувати маршрут для проходження усієї площини, необхідно прокласти лінію між лівим та правим краєм площини з певним проміжком, як показано на рисунку 7.1.

					ІА61.330БАК.005 ПЗ	А 41
З	А	№ докум.	Підп	Д		

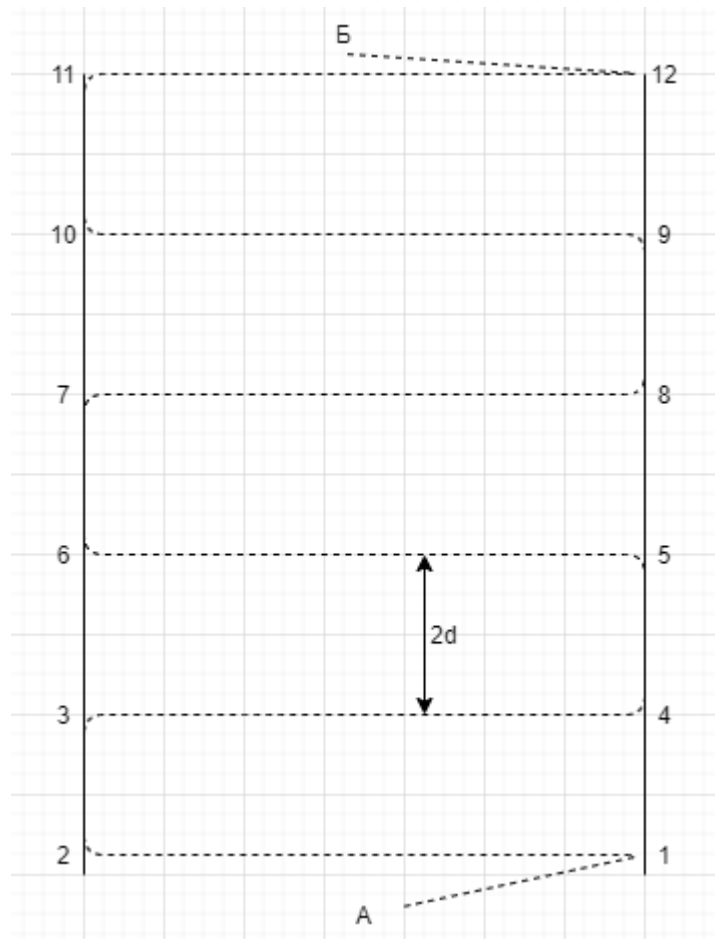


Рисунок 6.1 – Маршрут проходження площі

На рисунку 6.1 точка А – початкова точка руху системи, точка Б – кінцева точка руху. $2d$ – дві половини ширини огляду камери. Щоб обчислити половину ширини огляду камери, необхідно помножити дальність огляду (у даному випадку, дальність роботи сенсору LiDAR) на тангенс половини кута огляду. Оскільки кут огляду камери 160 градусів, краї можуть бути замилені і комп'ютерний зір не розпізнає сміття. Тому камера повинна бути налаштована

таким чином, щоб кут огляду був 90° , тоді половина кута – 45° . В такому випадку, половина ширини огляду камери:

$$8 * \operatorname{tg}(45^\circ) = 8\text{м.}$$

Тоді проміжок між лініями отриманого маршруту складе – 16 метрів. Цього достатньо, щоб сміття на відстані 8 метрів від системи попадало у видимість камери.

Далі контролер зберігає координати усіх отриманих точок маршруту із їх порядком, після чого прокладає маршрут між кожною точкою із обминанням перешкод за допомогою алгоритму хвильового трасування. Робота даного алгоритму складається із трьох основних етапів: ініціалізації, розповсюдження хвилі та відновлення шляху.

На етапі ініціалізації будується образ усіх комірок поля, та для кожної комірки створюються атрибути проходження/непроходження, зберігаються координати початкової та кінцевої комірки.

Наступним етапом від стартової комірки виконується крок у сусідню комірку, при цьому відбувається перевірка, чи можливо пройти дану комірку та чи не перевірялась вона раніше. Таким чином перевіряються усі вісім комірок навколо поточної, включаючи діагональні. При цьому в атрибуті комірки записується число, що дорівнює кількості кроків від стартової комірки до поточної. Кожна комірка, що перевіряється стає стартовою.

На етапі відновлення шляху перевірка комірок відбувається від фінішної комірки: навколо перевіряються усі комірки та обирається та, що має найменший атрибут проходження шляху. Далі таким чином перевіряється кожна комірка, що має атрибут на 1 менше поточної, а коли перевірка

доходить до стартової комірки – пройдений шлях зберігається. Отриманий шлях – найкоротший шлях до наступної точки з оминанням перешкод.

Приклад прокладеного шляху наведено на рисунку 6.2.

7	7		6	6	6	6	6	6	6	7	8
6	6		5	5	5	5	5	5	6	7	8
5	5	5	4	4	4	4	4	5	6	7	8
4	4	5	4	3	3	3	4			7	8
3					2	3	4	5	6	7	8
3	2	1	1	1	2	3	4	5			8
3	2	1	0	1	2	3	4	5			8
3	2	1	1	1	2	3	4	5	6	7	8

Рисунок 6.2 – Шлях, прокладений алгоритмом хвильового трасування [37]

За ширину комірки слід взяти довжину корпусу системи – 0.52м для того, щоб при оминанні перешкод, система не зачеплювала їх.

Блок 4. Пересування по маршруту. Коли маршрут побудовано, система може розпочинати рух по маршруту. Для цього контролер подає на драйвер електричних двигунів сигнал команду «вперед». Драйвер подає струм з акумуляторних батарей на обидва двигуни і система починає рухатися вперед. Обране шасі використовує танкову ходу, тому у випадку необхідності виконати поворот, на той двигун, що знаходиться зі сторони, у яку необхідно повернути подається команда «назад», а на інший двигун «вперед». Таким чином робот виконує поворот нульового радіусу. Тим часом контролер починає відміряти кут повороту, використовуючи дані з гіроскопу. Коли кут повороту збігається з кутом повороту на маршруті, маневр виконано і система продовжує рух далі.

Момент, коли необхідно виконати поворот, контролер обчислює завдяки даним з GPS модулю. Коли отримані з нього координати збігаються із

координатами повороту на маршруті, контролер дає на драйвер команду виконати поворот.

Блок 5. Розпізнавання сміття. Коли система пересувається по маршруту, настає етап пошуку та розпізнавання сміття. Камера вмикається, потокове зображення передається на контролер, де зображення обробляється та за допомогою комп'ютерного зору розпізнаються об'єкти, що є сміттям. На даному етапі визначається тип сміття та відбувається його сортування контролером у відповідну категорію. Процес розпізнання сміття продемонстровано на блок-схемі ІА61.330БАК.005 Д2 та детально описано у розділі 7.

Блок 6. Пересування системи до сміття. Оскільки сміття з великою імовірністю не знаходиться прямо на побудованому раніше маршруті, для його збору необхідно відхилитися від маршруту, попередньо зберігши координати початку руху до сміття.

На першому кроці задача полягає у тому, щоб обчислити відстань від кріплення маніпулятора до об'єкта, яка необхідна для того, щоб маніпулятор міг збирати сміття. Алгоритм можна розділити на дві частини: знаходження 2D положення об'єкту перед камерою та знаходження відстані від основи маніпулятора до об'єкту.

Перша частина алгоритму працює наступним чином. Спочатку отримується центр кадру. Потім обчислюється центр рамки знайденого об'єкту, що отримана алгоритмом, працюючим на основі згорткової нейромережі.

Коли координати обох центрів відомі, необхідно мінімізувати відстань між їх горизонтальними центрами. Для цього використовується поворот системи до об'єкту. Оскільки обране шасі здатне повертатися на місці, контролер дає двигуну, що знаходиться з протилежного до знайденого сміття боку, команду «вперед», а іншому двигуну, відповідно «назад». У момент,

					ІА61.330БАК.005 ПЗ	А 45
З	А	№ докум.	Підп	Д		

коли центр сміття збігся із горизонтальним центром кадру, на обидва двигуни подається команда «стоп».

Перша частина продемонстрована на рисунку 6.3, де d – горизонтальна, а h вертикальна відстань між центром кадру та центром рамки, якою обведений знайдений об'єкт.

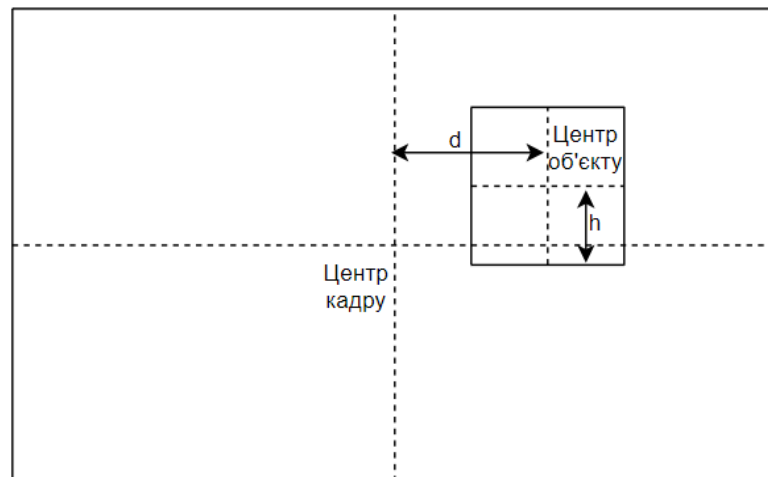


Рисунок 6.3 – Визначення 2D положення об'єкта перед камерою

Як тільки центри вирівнюються, наступна задача – перемістити робота на відстань, з якої він зможе зібрати сміття маніпулятором. На даному етапі сенсор LiDAR направлений на знайдене сміття, тому контролер відправляє запит на цей сенсор та отримує з нього дані щодо відстані до знайденого сміття. Далі контролер відправляє команду «вперед» на обидва двигуни і циклічно відправляє запити про відстань на сенсор LiDAR. У момент, коли відстань до сміття відповідає необхідній відстані для роботи маніпулятора, контролер зупиняє двигуни.

Блок 7. Керування маніпулятором. Наступним етапом контролер подає відповідні сигнали на сервоприводи, що у свою чергу рухають частини маніпулятора. Так, як сміття знаходиться на необхідній відстані перпендикулярно до маніпулятора, а контейнери для сміття знаходяться у

сталому положенні відносно маніпулятора, для підбору сміття і його складання у відповідний контейнер використовуються готові набори команд. Коли підібраний об'єкт поміщено у контейнер, маніпулятор повертається у вихідне положення.

Блок 8. Повернення на початкову позицію. Для продовження маршруту необхідно повернути систему на місце, з якого вона почала маршрут до сміття. Тому контролер відправляє на двигуни команди руху у різні сторони для розвороту системи та циклічно відправляє запит на гіроскоп, доки система не розвернеться на 180°. Далі контролер дає двигунам команду «вперед» та починає таким самим чином приймати дані з GPS модуля. Коли отримані координати збігаються із координатами, що були збережені у блоці 6, контролер зупиняє двигуни.

Блок 9. Рішення «Заданий маршрут пройдений». На даному етапі контролер, використовуючи дані з GPS модуля, звіряє їх із координатами кінцевої точки маршруту і приймає рішення щодо подальшої роботи. Якщо координати збігаються – маршрут пройдений, алгоритм переходить до блоку 10 та система повертається на вказані координати. В іншому випадку – маршрут не пройдений, наступним кроком алгоритму буде блок 2, де система або повертається на вказані координати за рішенням користувача, або продовжує роботу за описаним алгоритмом.

Блок 10. Повернення на вказані координати. Для повернення на вказані координати, контролер використовує описаний у блоці 3 алгоритм хвильового трасування для побудови прямого маршруту до вказаних координат з оминанням перешкод на шляху. Коли алгоритм побудовано, контролер керує двигунами, як описано у блоці 4 для досягнення кінцевої точки.

7 ДОДАТОК ДЛЯ ДИСТАНЦІЙНОГО КЕРУВАННЯ СИСТЕМОЮ

Для того, щоб мати можливість дистанційного відстеження стану системи, а також можливість дистанційно віддавати команди на повернення, змінювати кінцеву точку маршруту та ін., системі потрібен клієнт-серверний додаток.

Такий додаток має складатися із трьох основних частин: підпрограма на контролері самої системи, мобільний додаток для керування системою, а також серверна частина для зв'язку користувача з системою.

7.1 Підпрограма системи

Частиною програмного забезпечення системи є підпрограма, що відповідає за її зв'язок із сервером та використовує модуль зв'язку. Дана підпрограма із деякою періодичністю відправляє на сервер два види запитів: запит на отримання інформації з серверу та запити, що містять інформацію, що повинна бути відправлена на сервер.

Після кожного підбору сміття, додаток відсилає запити на сервер для перевірки, чи є команда повернення на вказані координати від користувача, після чого, приймає відповідне рішення щодо подальших дій. Також, якщо система увімкнена, але ще не працює – підпрограма очікує від сервера команду для початку роботи.

Для того, щоб користувач мав повну актуальну інформацію щодо стану системи, підпрограма періодично відправляє на сервер запити, що містять таку інформацію, як:

- а) текстову змінну, що містить у собі статус системи – працює, не працює чи закінчила роботу;
- б) числову змінну з рівнем заряду акумуляторів у відсотках;

					ІА61.330БАК.005 ПЗ	А 48
З	А	№ докум.	Підп	Д		

в) числову змінну, що містить координати системи, отримані з модулю GPS;

г) дві числові змінні із кількістю зібраного сміття кожного типу;

г) змінну часу, що містить час відправки інформації.

Уся перелічена інформація відправляється одним пакетом одночасно, що дає можливість користувачу розуміти повну інформацію на момент часу, коли її було відправлено.

7.2 Серверна частина

Серверна частина програми необхідна для непрямого зв'язку користувача із системою для отримання актуальної інформації, а також відправлення команди повернення системи, якщо це необхідно. Для того, щоб обидві частини могли отримувати інформацію, серверна частина містить базу даних, у якій зберігаються змінні, що відправлені з системи, а також команди, відправлені користувачем. У момент часу, коли на сервер приходить запит від системи для перевірки наявності команд, він відправляє системі булеву змінну команди з бази даних, яка приймає логічну одиницю у тому випадку, якщо користувач давав таку команду, а також змінну із координатами, на які система має повернутися.

7.3 Мобільний додаток

Оскільки система має працювати у різній місцевості, керуюча нею людина може знаходитися не на одному місці у приміщенні, а, наприклад, на вулиці. Тому, найкращим варіантом для керування буде мобільний додаток. Оскільки наразі майже всі мобільні телефони працюють на базі операційних

					ІА61.330БАК.005 ПЗ	А 49
З	А	№ докум.	Підп	Д		

систем IOS та Android, додаток повинен працювати на обох цих системах та бути написаним мовами програмування Swift та Java відповідно.

З певною періодичністю, що дорівнює періодичності відправки інформації з системи на сервер, даний додаток посилає запит на сервер та отримує увесь пакет інформації, що відправляє система.

Оскільки основна ціль додатку – відстеження стану системи – у додатку повинна бути мапа, на якій користувач зможе візуально подивитися, де знаходиться система. Для цього додаток використовує Google Maps api, що дозволяє інтегрувати у програму мапу місцевості із готовою логікою взаємодії користувача із нею. Для відображення системи на мапі використовуються координати із інформації, що була відправлена системою. Дані координати відображаються на мапі у вигляді точки.

Окрім мапи, додаток виводить інші дані, отримані з системи користувачу у вигляді таблиці.

Таким чином, користувач може подивитися рівень заряду акумуляторів для розуміння того, скільки часу приблизно ще може працювати система, кількість зібраного сміття різного типу для розуміння наповненості баків, а також стан роботи системи.

Також у додатку містяться дві кнопки для керування:

- перша кнопка – для повернення системи на вказану позицію – для того, щоб повернути систему раніше, ніж вона закінчить роботу, якщо на це з'явилися причини. Після її натискання з'являється мапа, на якій користувач обирає точку, куди повинна повернутися система. Далі координати обраної точки відправляються на сервер, де зберігаються, доки система їх не отримає;
- друга кнопка – відправляє системі команду розпочати роботу. Дана кнопка активна тільки тоді, коли система ще не працює.

8 РОЗРОБКА НЕЙРОМЕРЕЖІ РОЗПІЗНАВАННЯ СМІТТЯ

Існує декілька традиційних методів, за допомогою яких автономна система збирання сміття може виявити сміття, але жоден з них не є надійним, а точність на порядок нижча, ніж у людей. Для досягнення точності, наближеної до людської, система повинна використовувати метод машинного навчання. Такі методи вже майже досягли точності, як у людини, а в деяких випадках, навіть перевершили людей.

Для рішення задачі виявлення об'єктів, мобільна автономна система збирання сміття використовує згорткову нейронну мережу. Згорткові нейронні мережі – це клас глибинних штучних нейронних мереж, що широко використовується для класифікації зображень, виявлення об'єктів на зображеннях та в інших задачах комп'ютерного зору. Концепцією для згорткових нейронних мереж став паттерн зв'язку нейронів зорової кори людини та тварин. Ще одна причина, чому згорткова нейромережа є найбільш переважною для системи збирання сміття – вона не потребує багато рендерингу для розпізнання об'єкта на кадрі. Загальна різниця між згортковими нейромережами та традиційними алгоритмами полягає в тому, що традиційні алгоритми потребують розроблених вручну фільтрів, в той час, як згорткові мережі тренуються, вивчаючи особливості об'єктів багато разів для швидкого та точного їх подальшого розпізнання. Це робить нейромережі більш переважними за інші алгоритми.

Для виявлення сміття на зображенні автономна система збирання сміття використовує згорткову нейромережу для отримання рамки на зображенні навколо сміття. Саме тут система виконує задачу розпізнання об'єкта. Виявлення об'єктів на кадрі є ідентифікацією екземплярів об'єктів відповідного класу (таких, як скляна або пластикова пляшка) в зображеннях та відео у цифровому форматі. Алгоритм виявлення об'єкту дозволяє системі

ідентифікувати місця на зображенні чи відео, де знаходиться об'єкт. Для створення алгоритму виявлення об'єктів використовується попередньо натренована модель, що дозволяє системі ідентифікувати об'єкти, що лежать перед роботом. Після цього програма загрузається в пам'ять контролера системи.

Такий підхід до розпізнання сміття має декілька обмежень. Система не може розпізнати об'єкти, на які модель не натренована. Для демонстрації система натренована розпізнавати пластикові та скляні пляшки. Для реального використання потрібно навчити модель розпізнавати екземпляри найбільш частих видів вуличного сміття (таких, як пакети, недопалки і т.д.)

На рисунку 8.1 приведено власну реалізацію програми, що розпізнає сміття на прикладі пластикових та скляних пляшок.

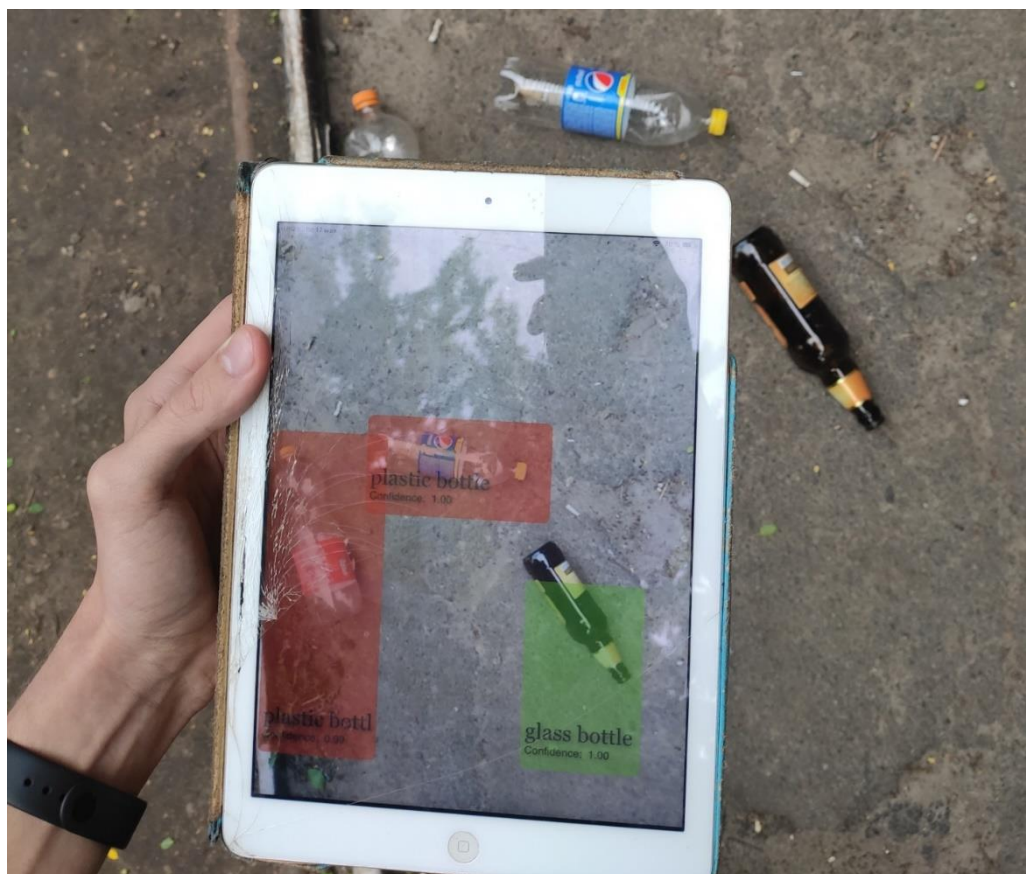


Рисунок 8.1 – Реалізація розпізнавання сміття

8.1 Тренування моделі комп'ютерного зору

Для реалізації розпізнавання пляшок було створено вибірку з 139 фотографій пластикових пляшок та 137 фотографій скляних пляшок. Для вибірки були взяті пляшки різного об'єму, форми та кольору для підвищення точності розпізнавання різних пляшок. Фотографії були зроблені з різним фоном, ракурсом, освітленням та в різних умовах.

Після того, як вибірка була готова, було розпочато процес створення анотацій на фотографіях.

Створення анотацій було виконано за допомогою сервісу cloud.annotations.ai. Для їх створення у даному сервісі необхідно обвести рамкою кожен об'єкт на кадрі, що необхідний для тренування, та додати назву для кожного обведеного об'єкту. Різні екземпляри одного об'єкту, такі, як, наприклад, скляна пляшка, повинні мати однакові назви. Увесь описаний алгоритм дій необхідно повторити для кожної фотографії. Процес виділення об'єктів на фотографіях продемонстровано на рисунку 8.2.

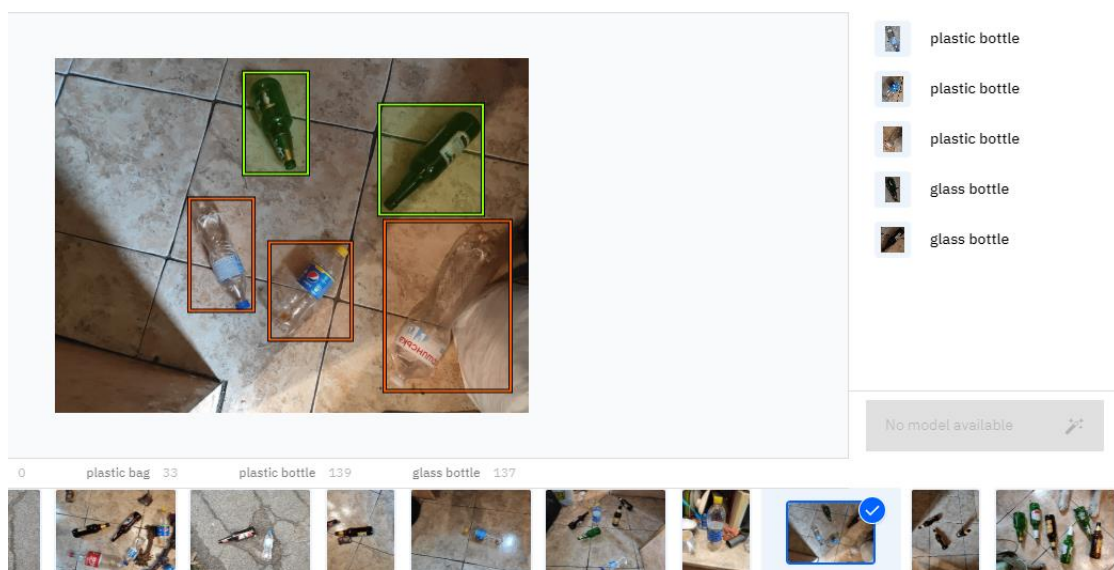


Рисунок 8.2 – Процес виділення об'єктів на фотографіях

Після того, як об'єкти на кожній фотографії підписані, сервіс створює файл із анотаціями та зберігає його на комп'ютері. Кожна анотація містить інформацію щодо назви фотографії, на якій є об'єкт, назви самого об'єкту, координат об'єкту на кадрі, ширини та висоти об'єкту. Приклад даних файлу з анотаціями зображено на рисунку 8.3.

```
[{"image": "f12f78a8-a3d9-4e9a-272-47da3ae798a8.jpg",
"annotations": [{"label": "plastic bag",
"coordinates": {"x": 734, "y": 615, "width": 503, "height": 834}}]},

{"image": "0922bc83-0bde-4b7e-8448-e1a9b9cb600c.jpg",
"annotations": [{"label": "plastic bag",
"coordinates": {"x": 636, "y": 515, "width": 463, "height": 770}}]},

{"image": "0403e071-314e-4a5e-85fa-c5738fb24e8b.jpg",
"annotations": [{"label": "plastic bag",
"coordinates": {"x": 721, "y": 596, "width": 361, "height": 399}}]},

{"image": "8e4336f2-eb8c-4837-b335-ce8577c56548.jpg",
"annotations": [{"label": "plastic bag",
"coordinates": {"x": 667, "y": 571, "width": 483, "height": 672}}]},

{"image": "3fa02911-aa67-4305-8509-91c7eed361d4.jpg",
"annotations": [{"label": "plastic bottle",
"coordinates": {"x": 905, "y": 611, "width": 493, "height": 260}}]},

{"image": "b8af9ccf-4a8f-47ff-8c70-a4280112d4e8.jpg",
"annotations": [{"label": "plastic bottle",
"coordinates": {"x": 831, "y": 643, "width": 439, "height": 304}}]},
```

Рисунок 8.3 – Дані файлу з анотаціями

Після того, як файл та вибірка готові – було розпочато тренування моделі комп'ютерного зору за допомогою бібліотеки Core ML. Саме на цьому етапі застосовується машинне навчання. Алгоритми вивчають особливості об'єктів одного екземпляру на фотографіях, такі як контури об'єктів, їх кольори, прозорість, блискучість та інше. У процесі вивчення об'єктів, алгоритми створюють зв'язки, що подібні зв'язкам нейронів в корі ока людини

та створюють на їх основі модель комп'ютерного зору. Завдяки збереженим зв'язкам, контролер здатен розпізнати об'єкт на зображенні дуже швидко, не потребуючи потужного заліза.

8.2 Реалізація додатку для розпізнавання сміття

Коли модель комп'ютерного зору готова, необхідно реалізувати програму, що, використовуючи дану модель, буде розпізнавати сміття на зображенні з камери.

Для демонстрації роботи розпізнавання, додаток було реалізовано мовою Swift у середовищі розробки Xcode.

Першим кроком було створено проект та завантажено у нього натренованої раніше моделі комп'ютерного зору. Далі було створено клас, у якому зберігається модель та методи асинхронного виклику нейронної мережі на виконання. На виході з даного класу створюються масиви даних MultiArray.

Модель використовує у якості виходу три шари для розбиття зображення на різну сітку, розміри осередків цих сіток мають такі значення: 8, 16 і 32. Припустимо на вході є зображення розміром 416x416 пікселів, тоді вихідні матриці (сітки) матимуть розмір 52x52, 26x26 і 13x13 ($416/8 = 52$, $416/16 = 26$ і $416/32 = 13$).

Після запуску проекту, на виході із створеного раніше класу створюються три об'єкти класу MLMultiArray. У властивості shape одного з отриманих об'єктів, вказані наступні дані [1,1,21,13,13]. З цих даних видно, що розмірність матриці – 13x13. Число 21 – розмірність матриці, так як кожен елемент матриці – вектор. Дане число обчислюється за формулою 8.1

$$B \times (5 + C), \quad (8.1)$$

					ІА61.330БАК.005 ПЗ	55 ^А
З	А	№ докум.	Підп	Д		

де B – кількість Bounding box в комірці. Для даної моделі у кожній комірці їх 3;

C – кількість об'єктів, на які натренована для розпізнавання модель. У даному випадку це 2 об'єкти – пластикові та скляні пляшки;

4 – сума коефіцієнту вірогідності наявності об'єкту, координат X , Y , а також ширини і висоти обмежувального прямокутника навколо передбаченого об'єкту. На рисунку 8.4 зображене схематичне представлення даного вектору.

Для даної нейромережі, натренованої на 2 об'єктах, для кожної комірки сітки передбачається 3 обмежувальні рамки, для кожної з них – 2 вірогідності об'єктів, коефіцієнт вірогідності об'єкту та 4 числа, що відповідають за розмір та положення даної рамки. Тож $3 \times (5 + 2) = 21$.

Після отримання цих даних, виконується функція Softmax, що перетворює вектор x розмірності K у вектор тієї ж розмірності, де кожна координата x_i отриманого вектору представлена числом у інтервалі $[0,1]$ та сума координат рівна 1. Обчислюється дана функція за формулою 8.2 .

$$\sigma(x)_i = \frac{e^{x_i}}{\sum_{k=1}^K e^{x_k}}, \quad (8.2)$$

де K – розмірність вектора;

$\sigma(\hat{x})$ – функція сигмоїди.

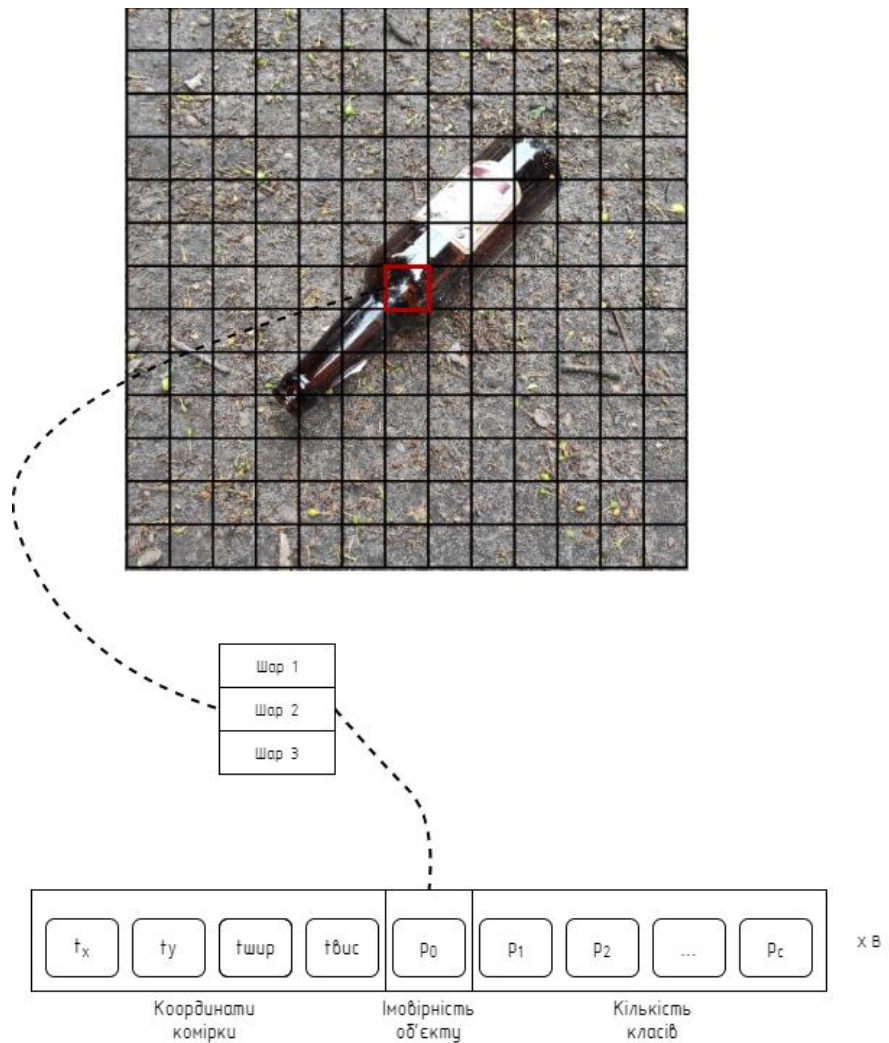


Рисунок 8.4 – схематичне представлення вектору

Для отримання координат та розмірів обмежувальної рамки, необхідно використати формули 8.3, 8.4, 8.5, 8.6

$$x = \sigma(\hat{x}) + C_x, \quad (8.3)$$

$$y = \sigma(\hat{y}) + C_y, \quad (8.4)$$

$$w = p_w e^{\hat{w}}, \quad (8.5)$$

$$h = p_h e^{\hat{h}}, \quad (8.6)$$

де x, y – x та y координати відповідно;

w – ширина рамки;

h – висота рамки;

$\sigma(\hat{x})$ – функція сигмоїди;

p_w, p_h – значення якорів, що були визначені під час тренування та збережені в файлі.

Після того, як координати визначені, можна створювати рамку навколо об'єкту. Для цього можна використати графічні бібліотеки мови програмування.

Увесь алгоритм перелічених дій циклічно виконується для обробки кожного кадру, отриманого з камери для того, щоб у випадку, коли об'єкт перемістився відносно камери, дані щодо його координат та розмірів на кадрі одразу оновилися.

					ІА61.330БАК.005 ПЗ	А 58
З	А	№ докум.	Підп	Д		

ВИСНОВКИ

В дипломному проєкті була розроблена мобільна автономна система збору та сортування сміття. В процесі розробки було розглянуто існуючі концепції систем автоматичного збирання сміття, проаналізовано їх недоліки та переваги, розглянуто їх принцип роботи та розроблено власну концепцію з можливими варіантами існуючих компонентів.

Розроблена система передбачає можливість роботи на вулицях міст та передмість, на бездоріжжі, економлячи при цьому людські ресурси та фінанси, що витрачаються на прибирання.

Також було виявлено, що комерційних аналогів описаного проєкту на даний момент немає, а існуючі концепти на даний момент не закінчені. У системи, описаній у дипломному проєкті, наявні такі переваги перед аналогами, як сортування зібраного сміття, побудова маршруту по території, яку необхідно прибрати з урахуванням перешкод, можливість відстеження стану та місцезнаходження системи через додаток, а також можливість дистанційного повернення системи до того, як вона закінчить роботу.

Були підібрані та описані компоненти, необхідні для роботи системи. Були проаналізовані інтерфейси входів та виходів із метою переконатися, що компоненти можуть працювати між собою. Була обчислена потужність, що необхідна для живлення усієї системи, підібрано акумуляторні батареї та описано їх розміщення на корпусі.

У процесі розробки дипломного проєкту були побудовані структурні та функціональні схеми, що включають в себе блоки компонентів системи, що відповідають за різну діяльність. Також була побудована блок-схема алгоритму роботи системи, а також блок-схема розпізнавання сміття, що

					ІА61.330БАК.005 ПЗ	59 ^А
З	А	№ докум.	Підп	Д		

демонструє алгоритм роботи комп'ютерного зору з готовою моделлю комп'ютерного зору, що була натренована заздалегідь.

Був модифікований алгоритм побудови маршруту з урахуванням перешкод таким чином, щоб маршрут охоплював усю площину заданої території з урахуванням реальних параметрів компонентів, таких, як кут огляду камери та габарити корпусу системи.

Було натреновано модель комп'ютерного зору для розпізнавання пластикових та скляних пляшок, для чого було створено вибірку з 139 фотографій пластикових пляшок та 137 фотографій скляних пляшок, були створені анотації та розпочато тренування програмою Create ML.

Був створений додаток для розпізнавання пляшок на основі натренованої моделі.

Створена модель комп'ютерного зору може бути використана на обраному для системи контролері, а додаток для розпізнавання може бути портований для роботи на контролері.

Описана система має простір для вдосконалення у майбутньому, наприклад, може бути підібрана краща камера, або може бути створений каркас для захисту компонентів від дощу.

					ІА61.330БАК.005 ПЗ	А 60
З	А	№ докум.	Підп	Д		

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Робот з машинним навчанням для автоматичного підбору сміття з трави.
URL: <https://arxiv.org/ftp/arxiv/papers/1904/1904.13034.pdf>
2. Робот для збирання сміття на пляжі із використання бездротових технологій. URL:
<http://ipcbee.com/vol66/019-IEEA2014-A044.pdf>
3. Одометр. URL:
<https://superdvigatel.com/ustrojstvo-avtomobilya/datchiki/datchik-holla.html>
4. Зображення одометру. URL:
https://superdvigatel.com/wp-content/uploads/2018/06/011e88ef4a8328e08be9d913808b8290_XL.jpg
5. Інерціальний вимірювальний модуль. URL:
<https://www.analog.com/ru/products/adis16500.html#product-overview>
6. LiDAR. URL:
<https://ru.wikipedia.org/wiki/%D0%9B%D0%B8%D0%B4%D0%B0%D1%80>
7. Електродвигуни. URL
<https://engineering-solutions.ru/motorcontrol/motor/>
8. Ходова частина. URL:
http://vex.examen-technolab.ru/lessons/unit_9_drivetrain_design/93/
9. Зображення однобортного повороту. URL:
http://vex.examen-technolab.ru/images/u/curr/block_9/9_5_2.png
10. GPIO. URL:
<https://raspberrypi.ru/43-gpio-dlya-chaynikov-chast-1>
11. CSI. URL:
https://en.wikipedia.org/wiki/Camera_Serial_Interface

					ІА61.330БАК.005 ПЗ	А
						61
З	А	№ докум.	Підп	Д		

12. Драйвер двигунів. URL:

https://myrobot.ru/stepbystep/el_driver.php

13. Raspberry pi 4 model b. URL:

https://miniboard.com.ua/boards/854-raspberry-pi-4-model-b.html?gclid=CjwKCAjw5vz2BRAtEiwAbcVILy9Wz733ZKU7kDBwJ3eFZ0OmDvGE16ybodG9nGx2gscVuQMtb0lccxoCQQIQAvD_BwE

14. Зображення Raspberry pi. URL:

https://miniboard.com.ua/5619-thickbox_default/raspberry-pi-4-model-b.jpg

15. Камера. URL:

https://rozetka.com.ua/152275593/p152275593/?gclid=CjwKCAjwztL2BRATEiwAvnALcrMqIsmgZprgLFPJ2dBc-Oskik2jZEK2e5W-5lbHGVGUXSATW-bGxoC3OYQAvD_BwE

16. Зображення камери:

https://i2.rozetka.ua/goods/15386929/152275593_images_15386929461.jpg

17. Інерціальний вимірювальний модуль. URL:

<https://www.analog.com/ru/products/adis16500.html#product-overview>

18. Зображення інерціального вимірювального модулю. URL:

https://www.analog.com/-/media/analog/en/evaluation-board-images/adis16500_pcbzangle-web.gif?la=ru&h=270&thn=1&hash=E2090FD2BFB47F5B65BD2EB95C89BD6A

19. LiDAR. URL:

https://aliexpress.ru/item/32852024277.html?spm=a2g0o.productlist.0.0.7dd42020ykn6NW&algo_pvid=00315420-0915-4d00-900a-535544c9a419&algo_expid=00315420-0915-4d00-900a-535544c9a419-0&btsid=0b0a0ae215903528562914202e8c36&ws_ab_test=searchweb0_0,s_earchweb201602_,searchweb201603

					ІА61.330БАК.005 ПЗ	62 ^А
З	А	№ докум.	Підп	Д		

20. Зображення LiDAR. URL::

<https://ae01.alicdn.com/kf/He4ce6c689a5947e8b29e40cac2b5627af.jpg>

21. Модуль зв'язку. URL:

<https://arduino.ua/prod3020-modyl-svyazi-4g3g2gsgmgsrsgnss-dlya-raspberry-pi>

22. Зображення модулю зв'язку. URL:

https://arduino.ua/products_pictures/large_DPC363-5.jpg

23. Ходова частина. URL:

<https://1topshop.org.ua/p999399195-robot-avto-shassi.html>

24.Зображення ходової частини. URL:

https://images.ua.prom.st/2303132061_w640_h640_2303132061.jpg

25. Маніпулятор. URL:

<https://aliexpress.ru/item/32876306634.html?channel=twinner>

26. Зображення маніпулятора. URL:

https://ae01.alicdn.com/kf/HTB1PQwpf_qWBKNjSZFAq6ynSpXat.jpg

27. Акумулятори. URL:

https://rozetka.com.ua/179772804/p179772804/?gclid=CjwKCAjw5vz2BRAtEiwAbcVIL6DSaydIbDKpqzhe9RdyQDY0aSZhCZ0er0MQ9PMGg-2E0FVeKRMwExoCMwIQAvD_BwE

28. Зображення акумулятора. URL:

https://i8.rozetka.ua/goods/16633693/179772804_images_16633693343.jpg

29. Зображення батарейного блоку. URL:

https://bms.com.ua/91-thickbox_default/battery-holder-18650-1x.jpg

30. Зображення схеми підключення BMS 3s. URL:

https://bms.com.ua/73-thickbox_default/pcm-bms-3s-40a.jpg

31. BMS 3s. URL:

<https://bms.com.ua/pcm-bms/21-pcm-bms-3s-40a.html>

					ІА61.330БАК.005 ПЗ	А 63
З	А	№ докум.	Підп	Д		

32. Модуль заряджання акумулятора. URL:

https://rozetka.com.ua/195876395/p195876395/?gclid=CjwKCAjw5vz2BRAtEiwAbcVIL7_x749YqTOHvoa1kRA3-3mx3sZ838-36Pk0JEpUmNJlCNxILDMv_hoC220QAvD_BwE

33. Зображення модулю заряджання акумулятора. URL:

https://i2.rozetka.ua/goods/17368570/195876395_images_17368570689.jpg

34. Драйвер електродвигунів. URL:

https://fd24.com.ua/p1044865475-drajver-dvigatelya-kan.html?gclid=CjwKCAjw5vz2BRAtEiwAbcVIL-G4pj0jolUQ5rNnhrhgT3njDThtw0mFrCERY_xJz-yDVeGnZR2YmxoC3PoQAvD_BwE

35. Зображення драйверу електродвигунів. URL:

https://images.ua.prom.st/2254275120_w640_h640_drajver-dvigatelya-2-kan.jpg

36. Алгоритм трасування. URL:

https://ru.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC_%D0%9B%D0%B8

37. Зображення шляху, пройденого алгоритмом трасування:

https://upload.wikimedia.org/wikipedia/commons/4/4e/Lee_wave_8.png

38. Згорткові мережі. URL:

https://ru.wikipedia.org/wiki/%D0%A1%D0%B2%D1%91%D1%80%D1%82%D0%BE%D1%87%D0%BD%D0%B0%D1%8F_%D0%BD%D0%B5%D0%B9%D1%80%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B5%D1%82%D1%8C

39. CNN. URL:

<https://habr.com/ru/post/460869/>

40. Автономний робот для збирання сміття. URL:

					ІА61.330БАК.005 ПЗ	А 64
З	А	№ докум.	Підп	Д		

<http://article.sapub.org/10.5923.j.ijit.20170602.06.html>

41. Зображення автономного роботу для збирання сміття. URL:

http://article.sapub.org/image/10.5923.j.ijit.20170602.06_002.gif

42. Wasteshark. URL:

<https://newatlas.com/wasteshark-sea-plastic-waste-drone/45407/>

43.Зображення Wasteshark. URL:

<https://assets.newatlas.com/dims4/default/ea292a3/2147483647/strip/true/crop/1620x1080+0+0/resize/1200x800!/format/webp/quality/90/?url=http%3A%2F%2Fnewatlas-brightspot.s3.amazonaws.com%2Farchive%2Fwasteshark-1.jpeg>

44. Інформація про маніпулятори. URL:

[https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D0%BD%D0%B8%D0%BF%D1%83%D0%BB%D1%8F%D1%82%D0%BE%D1%80_\(%D0%B3%D1%80%D1%83%D0%B7%D0%BE%D0%BF%D0%BE%D0%B4%D1%8A%D1%91%D0%BC%D0%BD%D0%BE%D0%B5_%D1%83%D1%81%D1%82%D1%80%D0%BE%D0%B9%D1%81%D1%82%D0%B2%D0%BE\)](https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D0%BD%D0%B8%D0%BF%D1%83%D0%BB%D1%8F%D1%82%D0%BE%D1%80_(%D0%B3%D1%80%D1%83%D0%B7%D0%BE%D0%BF%D0%BE%D0%B4%D1%8A%D1%91%D0%BC%D0%BD%D0%BE%D0%B5_%D1%83%D1%81%D1%82%D1%80%D0%BE%D0%B9%D1%81%D1%82%D0%B2%D0%BE))

45. Інформація про розрахунок потужностей. URL:

<https://ru.wikihow.com/%D1%80%D0%B0%D1%81%D1%81%D1%87%D0%B8%D1%82%D0%B0%D1%82%D1%8C-%D0%BF%D0%BE%D1%82%D1%80%D0%B5%D0%B1%D0%BB%D1%8F%D0%B5%D0%BC%D1%83%D1%8E-%D0%BC%D0%BE%D1%89%D0%BD%D0%BE%D1%81%D1%82%D1%8C-%D1%8D%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D0%BF%D1%80%D0%B8%D0%B1%D0%BE%D1%80%D0%B0>

46. Створення анотацій. URL:

					ІА61.330БАК.005 ПЗ	А 65
З	А	№ докум.	Підп	Д		

<https://idaas.iam.ibm.com/idaas/mtfim/sps/authsvc?PolicyId=urn:ibm:security:authentication:asf:basicldapuser>

47. Тренування моделі. URL:

<https://www.hackingwithswift.com/books/ios-swiftui/training-a-model-with-create-ml>

					ІА61.330БАК.005 ПЗ	А 66
З	А	№ докум.	Підп	Д		